TILBURG UNIVERSITY

CLASSIFYING IN-HOSPITAL MYOCARDIAL INFARCTION COMPLICATION MORTALITY

A COMPARATIVE STUDY ON THE XGBOOST, TABNET, AND LONG-TERM-COGNITIVE-NETWORK ALGORITHMS

RENEE PEX

THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE IN DATA SCIENCE & SOCIETY
AT THE SCHOOL OF HUMANITIES AND DIGITAL SCIENCES
OF TILBURG UNIVERSITY

STUDENT NUMBER

2064669

COMMITTEE

dr. Gonzalo Napoles
dr. Grzegorz Chrupała

LOCATION

Tilburg University
School of Humanities and Digital Sciences
Department of Cognitive Science &
Artificial Intelligence
Tilburg, The Netherlands

DATE

December 4th, 2023

WORD COUNT

8800

ACKNOWLEDGMENTS

I want to thank my supervisor, Gonzalo Nápoles for his guidance during the supervision sessions throughout the semester. Furthermore, I would like to thank my supervision session group members for their support.

CLASSIFYING IN-HOSPITAL MYOCARDIAL INFARCTION COMPLICATION MORTALITY

A COMPARATIVE STUDY ON THE XGBOOST, TABNET, AND LONG-TERM-COGNITIVE-NETWORK ALGORITHMS

RENEE PEX

Abstract

Myocardial Infarction (MI) is one of the most challenging medical emergencies where the heart muscle begins to die due to a lack of blood flow. About 50% of the patients contract complications that lead to worsening of the disease or even death. Creating accurate machine and deep learning models that can predict lethal complications holds paramount importance for pre-emptive interventions. Conventional machine learning algorithm XGBoost is widely considered the current state-of-the-art for predictive modeling on tabular data. However, a shift from machine to deep learning techniques has been identified in the medical field. This study builds on the limited research on deep learning techniques for tabular data in the field of MI complication mortality classification. The MI dataset from the University of California machine learning repository was used to build predictive models with the Logistic Regression, XGBoost, TabNet, and Long-term Cognitive Network (LTCN) algorithms. Additionally, the study explores the effect of the Synthetic Minority Oversampling Technique (SMOTE) to address the class imbalance in the dataset. Furthermore, feature importance methods have identified the most prominent features. After evaluating model performances, the XGBoost combined with SMOTE model provided the best kappa score of 0.572, closely followed by the LTCN model without SMOTE, achieving a kappa score of 0.542. LTCN in combination with SMOTE yielded the highest ROC-AUC score of 0.820.

1 DATA SOURCE, ETHICS, CODE, AND TECHNOLOGY STATEMENT

1.1 *Source/Code/Ethics/Technology Statement Example*

The MI dataset has been acquired from the UCI machine learning repository through an online request. Work on this thesis did involve collecting data from human participants. The obtained data is anonymized. No new data was collected by the author. The original owner of the data and code used in this thesis retains ownership of the data and code during and after the completion of this thesis. However, the institution made the dataset publicly available for research. All the figures belong to the author. The thesis code can be accessed through the GitHub repository following the link [https://github.com/CodingPex]. In terms of writing, the author used assistance with the language of the paper. A generative language model named Grammarly was used to improve the author's original content, for paraphrasing, spell-checking, and grammar. Chat-GPT was used to help understand coding errors and to make small modifications to code. Finally, websites like kaggle.com, machinelearningmastery.com, stackoverflow.com, and towardsdatascience.com were consulted for coding examples and questions.

2 INTRODUCTION

This work will contribute to the literature on Myocardial Infarction (MI) and the potential of machine and deep learning techniques to assist in this medical emergency. Myocardial Infarction (MI) is a medical emergency where the heart muscle begins to die due to a lack of blood flow ("cleveland", 2023). MI is more commonly known as a heart attack and can be described as one of the most challenging problems of modern medicine (Golovenkin et al., 2020). In the US alone more than a million people suffer from MI every year.

About 50% of the patients in the acute and sub-acute periods contract complications that lead to the worsening of the disease or can even lead to mortality. Foreseeing these complications is one of the greatest challenges of this medical emergency. Therefore, using the potential of machine and deep learning techniques to help predict the mortality of potential complications, is a valuable task. Creating accurate models that can predict complications leading to death is important to carry out the necessary preventive measures.

TabNet and Long-Term Cognitive Network (LTCN) are relatively novel deep learning algorithms. These algorithms are designed explicitly for tabular data, which corresponds with the input data for this study. Notably, both algorithms claim to have high interpretability and explainability, which contributes to the social relevance of the study. Limited studies have tested deep learning techniques for tabular data to predict complications and/or mortality after an MI incidence. While TabNet has demonstrated promise in various classification and prediction tasks across diverse domains (Arik & Pfister, 2021), it has primarily been tested on large tabular datasets. It will be interesting to see how TabNet performs on the smaller dataset that is available for this binary classification study.

2.1 *Scientific Novelty*

Classifying complication mortality after an MI incidence has been studied extensively (see section 3). The widely adopted XGBoost algorithm is considered as the current state-of-the-art in studies with tabular data for this research domain. More recently, a transition from machine to deep learning techniques has been identified in the medical field (Garg & Mago, 2021; Sharifani & Amini, 2023). While tabular data is still the most common data type in real-world applications (Shwartz-Ziv & Armon, 2022), deep learning techniques for tabular data have generally trailed behind when compared to image, text, and audio data. More recently, deep learning algorithms such as TabNet and LTCN have been created specifically for tabular data. This study will test these deep learning algorithms and compare their performances to the XGBoost algorithm. Additionally, these algorithm's performances will all be compared to a baseline Logistic Regression model.

Furthermore, TabNet has mainly been tested on larger datasets (de Carvalho et al., 2023; Kim et al., 2023). It will be intriguing to see how it performs on the relatively small dataset that is available for this study. In contrast, LTCN, while less extensively tested, is computationally less expensive than TabNet and is not explicitly designed for large datasets. These characteristics might make LTCN more suitable for the classification task at hand. Both TabNet and LTCN claim to have higher interpretability than traditional black box models which contributes to the transparency of this study. The transparency will be further enhanced by providing a thorough error analysis, distinguishing this study from previous studies employing the same dataset. Finally, various feature importance methods will be tested to provide insights into the most prominent features, which contributes to the importance of explainable artificial intelligence in the medical field.

2.2 *Research Questions*

The main Research Question is as follows:

> *"To what extent do deep learning models perform compared to conventional machine learning models for classifying Myocardial Infarction (MI) complication mortality?"*

Several sub-questions have been derived to support the main research question and provide a basis for the study.

> *"RQ1: To what extent do the performance metrics of the TabNet, LTCN, and XGBoost models compare to the performance metrics of the baseline model for predicting MI complication mortality? "*

RQ1 provides an evaluation of all the tested algorithms. The models will be evaluated based on their kappa, ROC-AUC, F1, recall, and precision scores.

> *"RQ2: To what extent does the model performance differ when using SMOTE to address the class imbalance in the dataset?"*

RQ2 aims to evaluate the effect of applying SMOTE to the training data. The following performance metrics will be used to measure the effect: kappa, ROC-AUC, F1-Score, recall, and precision.

> *"RQ3: Which important features in the dataset can be identified through feature importance methods for the best-performing model?"*

RQ3 aims to identify the features in the dataset that contribute most to the predictive power of the various models. Shap, SKLearn's random feature permutation method, and the best-performing algorithm's built-in feature importance method will be tested. Through the pixel flipping experiment, the best-performing feature importance method will be chosen to identify the most prominent features.

2.3 *Main Findings*

After evaluating the model performances, XGBoost in conjunction with SMOTE yielded the best results in terms of the kappa score, achieving a score of 0.572. LTCN combined with SMOTE demonstrated superior performance in terms of ROC-AUC, achieving a score of 0.820. Unexpectedly, TabNet failed to outperform the baseline model. LTCN exhibited commendable performance without treating the class imbalance, achieving a kappa score of 0.542. Both XGBoost and LTCN were deemed most suitable for the classification task in this study. The application of SMOTE enhanced the model's ability to accurately classify instances belonging to the minority class. Additionally, this enhancement had minimal impact on misclassifying instances belonging to the majority class, particularly evident in the XGBoost model's performance. Further analysis revealed three prominent features contributing significantly to the classification task in this study: the electrocardiogram rhythm at the time of admission to the hospital, the use of calcium blockers in the intensive care unit, and the sex of the patient.

3 RELATED WORK

Machine learning has become an essential tool in the process of analyzing complex medical data. One literature review on this topic has concluded that a shift in artificial intelligence techniques is happening (Garg & Mago, 2021). This work states that deep learning techniques are taking precedence over conventional machine Learning techniques. This statement supports the plan to test the proposed deep learning algorithms for this work's classification task. This paper also identifies the importance of advancements in the field. In the US, 80% of healthcare spending is spent on chronic disease treatment. Furthermore, in China, 86% of disease-related deaths are connected to chronic diseases.

3.1 *Machine Learning and Myocardial Infarction*

Studies utilizing machine and deep learning techniques in the context of myocardial infarction have mainly focused on predicting the mortality rate among patients after experiencing an MI incidence (Cho et al., 2021). This study states that acute MI is the leading cause of death globally and explores the potential of machine learning to predict mortality after an MI incidence. SMOTE was applied to overcome class imbalance, which improved the model performance within Cho et al. (2021)'s study.

Furthermore, numerous studies have focused on predicting the likelihood of an MI incidence based on electrocardiogram (ECG) data (Kora et al., 2018) (Kora & Sri Rama Krishna, 2016) (Sharma et al., 2018). Additionally, Chakraborty et al. (2022)'s study provides a comprehensive review that systematically analyzed and discussed multiple research papers that employed machine and deep learning techniques to predict the likelihood of an MI incident (Chakraborty et al., 2022). The potential of machine and deep learning becomes apparent when assessing the high accuracy scores in the comparative table. However, several of the reviewed studies in this paper solely relied on accuracy as a performance metric. The absence of more comprehensive metrics such as recall and kappa, especially in the realm of medical data, can lead to inadequate assessments. Imbalanced datasets can provide high accuracy scores when solely predicting the majority class. This emphasizes the importance of including more comprehensive performance metrics since false negatives should be avoided.

When comparing the results with previous studies that utilized the same dataset, the most recent study by Newaz et al. (2023) has notably yielded the most compelling results. Newaz et al. (2023) tested various conventional machine learning algorithms for predicting complications of MI and XGBoost was the best-performing algorithm in terms of "ROC-AUC" score, which is a good performance measure when working with imbalanced data. The latest study demonstrates a notable increase of 0.12 in the ROC-AUC score when compared to the previous studies (Farah et al., 2022; Joshi et al., 2022; Reddy & Thangam, 2022). Interestingly, none of the aforementioned studies offer comprehensive insights into individual model performance concerning each class. Given the significant class imbalance present in the dataset, the absence of an error analysis presents a crucial research gap.

3.2 *TabNet and Long-Term Cognitive Network*

As mentioned in section 2.1, a shift from machine to deep learning techniques has been identified in the medical field. Both TabNet and LTCN are deep learning algorithms explicitly created for tabular data. TabNet has shown promise accros various datasets (Arik & Pfister, 2021). LTCN is inspired by fuzzy cognitive map-based models and designed for tabular data with well-defined features. In this FCM-like model, the weights are not constrained to a specific interval. Moreover, the tunable parameters within LTCN are determined using a non-synaptic backpropagation algorithm (Nápoles et al., 2021). The TabNet algorithm has been tested in two studies within the research domain of this study (de Carvalho et al., 2023; Kim et al., 2023). Both of these studies used different input data to create predictive models.

3.2.1 *TabNet*

de Carvalho et al. (2023) tested the TabNet algorithm for its ability to predict short-term outcomes after an MI incidence. Utilizing a relatively large dataset, the TabNet algorithm demonstrated superior performance, yielding an accuracy of 0.946 (de Carvalho et al., 2023). A second study by Kim et al. (2023), used a relatively extensive dataset of the Korea acute MI registry and observed TabNet's outperformance of conventional machine learning algorithms (Kim et al., 2023). These studies show the potential of the TabNet algorithm. An intriguing prospect for further exploration involves examining TabNet's ability to surpass conventional machine learning algorithms when applied to the smaller dataset available for this study.

3.2.2 *LTCN*

The LTCN algorithm is another deep learning algorithm created specifically for tabular data. Unlike TabNet, the LTCN algorithm has not yet been tested within the research domain of this study. LTCN is a recently proposed algorithm by Nápoles et al. (2021) that has undergone less extensive testing compared to TabNet. However, a comparative analysis suggests that LTCN might be better suited for the smaller dataset in this study. This proposition arises from LTCN's absence of explicit design for high dimensional or larger datasets and lower complexity, as is evident from its testing across datasets ranging from 846 to 10,922 instances. LTCN provides competitive performance when compared to state-of-the-art algorithms in terms of kappa score on the tested datasets (Napoles et al., 2022).

3.3 *Class imbalance*

As described by Chawla et al. (2002), real-world datasets often consist of a majority of "normal" cases and a smaller percentage of "abnormal" cases (Chawla et al., 2002). Class imbalances can lead classifiers to be biased towards the majority class. There are various ways to address class imbalances such as resampling techniques, cost functions, and class weights (Abd Elrahman & Abraham, 2013).

Newaz et al. (2023) explored several different methods to handle class imbalance in the MI dataset. Most notably, Newaz et al. (2023) proposed a new hybrid approach that combined sampling techniques with the cost-sensitive learning framework. Within this approach SMOTE was used as resampling approach as it provided the best performance (Newaz et al., 2023). Other variations of SMOTE like Edited Nearesst Neighbours (ENN), Tomek-links, and Adaptive Synthetic oversampling (ADASYN) were also tested. The previous studies that used the same MI dataset all failed to provide an error analysis, which prevents seeing the effect of class imbalance treatment for each class (Farah et al., 2022; Joshi et al., 2022; Newaz et al., 2023; Reddy & Thangam, 2022). Lastly, Rai and Chatterjee (2022) used a hybrid approach of SMOTE and Tomek link sampling methods to tackle class imbalance. This approach parallelly generates oversamples while maintaining data dissimilarity between minority and majority samples, and also balance the classes (Rai & Chatterjee, 2022).

3.4 *Feature importance methods*

Explainable artificial intelligence has emerged as an important research direction to provide insights and explanations for the behaviour of more complex machine and deep learning models. Within this research direction, feature importance techniques are one of the most popular approaches to provide more transparency and interpretability (Saarela & Jauhiainen, 2021). Explainable artificial intelligence is important when working with sensitive medical data, where transparency about how complex models work is crucial for full adoption (Mohanty & Mishra, 2022).

Local Interpretable Model-agnostic Explanations (LIME) and SHapley Additive exPlanations (SHAP) are two popular examples of post-hoc interpretable machine learning methods (Knapič et al., 2021). There are multiple studies in the healthcare domain that have used one or both of these methods to provide insights into prominent features and how they contributed to the predictions (Alabi et al., 2023; Lewin-Epstein et al., 2021; Prendin et al., 2023; Seki et al., 2021; Wu et al., 2023). Lai et al. (2019) conducted a comparison between feature importance techniques from built-in model mechanisms and post-hoc methods like SHAP and LIME. Their findings indicated that post-hoc methods tend to yield more similar prominent features more consistently (Lai et al., 2019).

3.5 *Research Gaps*

As inferred in section 3, a shift from machine to deep learning techniques has been identified in the medical field. Deep learning techniques for tabular data have not experienced the same improvements as other forms of data in recent years (Jang & Cho, 2019; Razzak et al., 2018). However, recently more attention has been given to the development of new algorithms with a deep learning architecture to compete with the current state-of-the-art for tabular data. The limited studies in the field of MI complication classification that have tested these deep learning techniques for tabular data introduce the first research gap.

Two of these deep learning algorithms for tabular data are TabNet and LTCN. TabNet has been tested extensively and has been able to outperform XGBoost in various fields (Arik & Pfister, 2021). One study by Nguyen and Byeon (2023) tested the TabNet algorithm for the prediction of out-of-hospital cardiac arrest survival. In that study, TabNet outperformed the XGBoost algorithm with a "ROC-AUC" score of 0.9934 (Nguyen & Byeon, 2023). While Nguyen and Byeon (2023) predicted out-of-hospital cardiac arrest, this study will aim to predict if patients contract a fatal complication before leaving the hospital.

Given TabNet's purported suitability for large tabular datasets, its performance on the comparatively smaller dataset available for this study presents an interesting research gap. Conversely, the LTCN algorithm is not necessarily designed for larger datasets (Napoles et al., 2022). Thus, while TabNet has been tested more extensively with success, LTCN could be more suitable for this particular study while also offering computational efficiency. Both TabNet and LTCN claim to have higher interpretability than the traditional black box models which contributes to the transparency of the study.

Finally, multiple feature importance methods will be tested during the feature importance analysis to determine the best method and most prominent features. Identifying these features will provide social relevance and can help provide insights for future research. Furthermore, previous studies failed to provide an error analysis of the predictions made by their models. This study will provide a transparent overview in the form of confusion matrices, supplemented by recall, precision and F1-scores.

4 METHOD

The Logistic Regression algorithm was used to build a baseline model for this study. Furthermore, the XGBoost model was identified as the current state-of-the-art in MI classification studies with tabular data. Lastly, the TabNet and LTCN algorithms were used to build deep learning models for tabular data. These algorithms will be explained and justified in this section. Figure 1 provides a flowchart of the data science pipeline in this study.

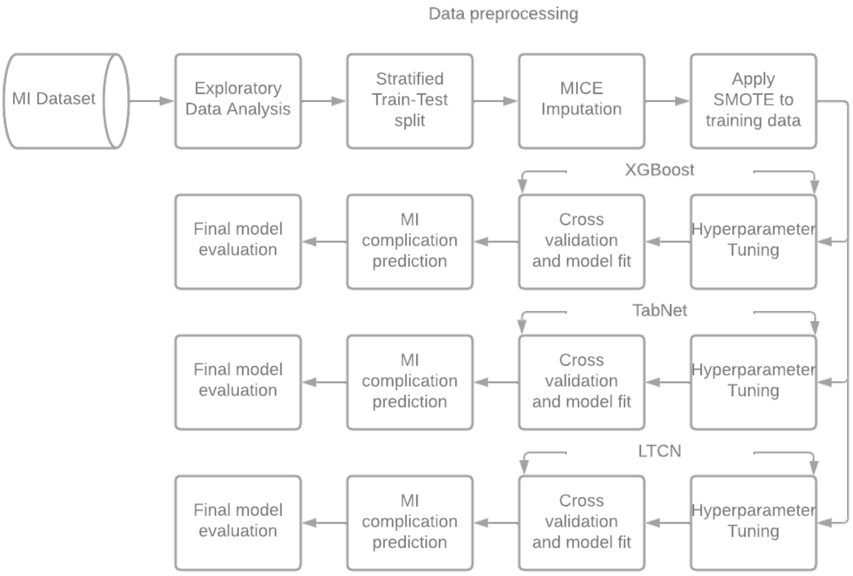

Data preprocessing
MI Dataset
Exploratory Data Analysis
Stratified Train-Test split
MICE Imputation
Apply SMOTE to training data
XGBoost
Final model evaluation
MI complication prediction
Cross validation and model fit
Hyperparameter Tuning
TabNet
Final model evaluation
MI complication prediction
Cross validation and model fit
Hyperparameter Tuning
LTCN
Final model evaluation
MI complication prediction
Cross validation and model fit
Hyperparameter Tuning

Figure 1: Data Science Pipeline

4.1 *Baseline Model*

The Logistic Regression algorithm was used to build the baseline model for this study. Logistic Regression is a common machine learning algorithm used in binary classification tasks across fields, including medical applications (Austin & Tu, 2004; Tsien et al., 1998). The convenience and low computational cost of the algorithm make it an ideal baseline model. Hyperparameter optimization was carried out using GridSearchCV, with the hyperparameter grid specified in table 1.

Table 1: The selected Hyperparameter grid for the baseline model

Hyperparameters	Default Value	Parameter Grid
C	1	[0.001, 0.01, 0.1, 1, 10, 100, 1000]
Solver	"lbfgs"	[liblinear, newton-cg] [lbfgs, sag, saga]

The "C" hyperparameter specifies the regularization of the model, where a smaller value specifies a stronger regularization. The "Solver" hyperparameter specifies the algorithm to be used in the optimization process (Pedregosa, Varoquaux, Gramfort, Michel, Thirion, Grisel, Blondel, Prettenhofer, Weiss, Dubourg, et al., 2011a).

4.2 *XGBoost*

XGBoost, a machine learning algorithm inspired by tree boosting, is extensively utilized across diverse research domains. It is widely acknowledged as the current state-of-the-art for tabular data (Chen & Guestrin, 2016b). XGBoost is an ensemble gradient boosting technique that builds multiple models which are then combined to produce superior results. XGBoost creates a more generalizable model through optimization of specific loss functions and the application of various regularization techniques, which mitigates the risk of overfitting (Chen & Guestrin, 2016b).

4.2.1 *XGBoost Hyperparameter Tuning*

Hyperparameter tuning is imperative for optimizing the XGBoost model, as emphasized in the work by Putatunda and Rama (2018) (Putatunda & Rama, 2018). The XGBoost documentation page was consulted to specify a hyperparameter grid. This hyperparameter grid was subsequently employed as input for the GridSearchCV technique, facilitating the identification of the optimal hyperparameters. The XGBoost hyperparameter grid utilized in this study is specified in table 2.

The various parameters are carefully chosen to find a balance in model complexity and performance. In general, lower values for subsample, colsample, max depth, and n estimators prevent overfitting. While a larger value for Min child weight and Gamma makes the model more conservative. Finally, the learning rate affects the time needed for the model to learn

Table 2: The selected Hyperparameter grid for the XGBoost model including the optimal value after Grid Search Cross Validation

Hyperparameters	Default Value	Parameter Grid	Best Value
Min child weight	1	[1, 5, 10]	0.5
Gamma	0	[0.5, 1, 1.5, 2, 5]	1
subsample	1	[0.6, 0.8, 1.0]	0.8
colsample bytree	1	[0.6, 0.7, 0.8, 0.9]	0.9
max depth	6	[3, 4, 5]	6
n estimators	100	[100, 200, 400]	200
learning rate	0.3	[0.01, 0.1, 0.2]	0.1

and is also used as a measure to make a model more conservative (Chen & Guestrin, 2016a).

4.3 *TabNet*

4.3.1 *TabNet's Architecture*

The most recent advancements in the field of Artificial Intelligence have mainly attributed to deep learning techniques for image, video, text, and audio data. Conventional machine learning algorithms are still widely considered superior to deep learning algorithms for tabular data (Shwartz-Ziv & Armon, 2022). In recent years however, more attention has been given to the development of deep learning algorithms for tabular data, leading to the introduction of TabNet by the Google team in 2019. TabNet's architecture operates more like a Deep Neural Network and claims to have the interpretability of traditional tree models (Arik & Pfister, 2021).

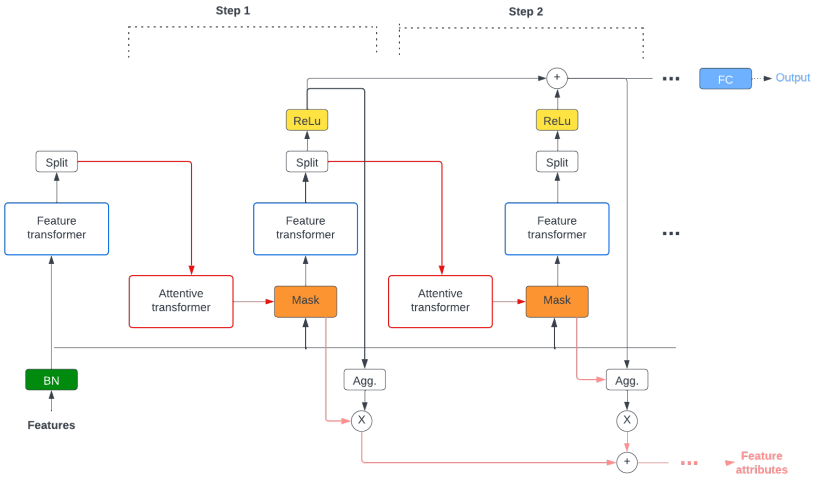


Figure 2: TabNet Encoder Architecture

Figure 2 shows the architecture of the TabNet algorithm. TabNet is made up of a feature transformer that extracts the features that pass through the algorithm. After passing through the feature transformer the attentive transformer is used for feature selection. TabNet optimizes the learning capacity and makes the model parameter efficient through sparse feature selection of the most prominent features (Arik & Pfister, 2021).

Figure 3 shows how the feature transformer works within the TabNet algorithm. At each decision step, the input data passes through a Fully Connected Layer (FC), Batch Normalization Layer (BN), and a Gated Linear Unit (GLU). Eventually, the decision steps are normalized with $\sqrt{0.5}$ to stabilize the learning and variance across the model (Gehring et al., 2017).

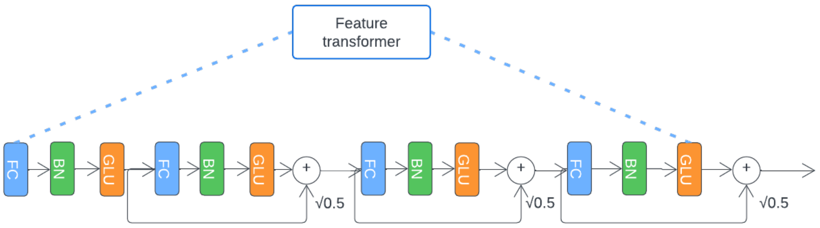


Figure 3: TabNet Feature Transformer

Figure 4 shows the function of the attentive transformer within the TabNet algorithm. TabNet uses a learnable mask based on prior decision steps to select the most prominent features. In this study, the sparsemax normalization was used for feature selection. The formula for the sparsemax mask is specified below.

$$M[i] = sparsemax(P[i-1] * h_i(a[i-1])) \tag{1}$$

P[i-1] represents the prior scale term that specifies how much a feature has been used at prior decision steps. At P[0], all features are initialized as ones since they have not passed through any decision steps yet. Later, unused features corresponding values are set as zeros. a[i-1] represents the processed feature information from the preceding step. h_i is a trainable function that can be trained through the Fully Connected and Batch Normalization layer seen in Figure 4.

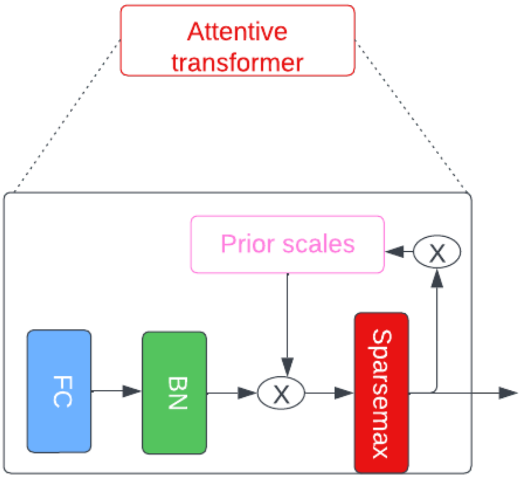


Figure 4: TabNet Attentive Transformer

Batch normalization is applied to the numerical and categorical features of the input data. Then, the same amount of N-dimensional features are passed to each decision step. TabNet is a sequential algorithm which means that at each decision step, the model takes the processed information of the $(i-1)^{th}$ step for feature selection. Subsequently, the processed feature representation is aggregated in the decision process (Arik & Pfister, 2021).

4.3.2 *TabNet Hyperparameter Tuning*

The TabNet documentation was consulted to form the hyperparameter grid. GridSearhCV was used to find the optimal hyperparameter combination within the specified parameter grid. The predetermined parameter grid can be seen in figure 3.

Table 3: The selected Hyperparameter grid for the TabNet model and the optimal value of the best performing model after GridSearchCV

Hyperparameters	Default Value	Parameter Grid	Best Value
N-steps	3	[3,5,7,9]	3
Gamma	1.3	[1,1.3,1.5,2]	1.3
N-independent GLU	2	[1,2,3,4]	1
N-shared GLU	2	[1,2,3,4]	1
Momentum	0.02	[0.02,0.05,0.1,0.4]	0.02

The N-steps parameter specifies the number of steps which can range from 3 to 10. Gamma is a coefficient for feature re-usage in the masks. The Gamma hyperparameter can range from 1 to 2, where a value closer to 1 means less correlation between layers during mask selection. Finally, the momentum hyperparameter for batch normalization typically ranges from 0.01 to 0.4 (Arik & Pfister, 2021).

4.4 *Long Term Cognitive Network (LTCN)*

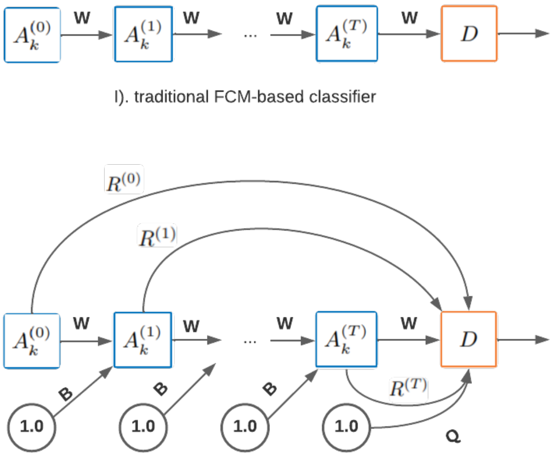
The relatively new LTCN algorithm proposed by Napoles et al. (2022) will be tested in this study as an alternative Deep Learning technique to TabNet. LTCN is a recurrence-aware model that is designed for explainable pattern classification. The LTCN-based algorithm introduces a quasi-nonlinear reasoning rule. This rule incorporates a nonlinear coefficient that controls the extent of the transfer function's impact on the neuron's initial activation value. LTCN can be seen as a type of recurrent neural network

(RNN) (Napoles et al., 2022). The target labels were one-hot-encoded to be compatible with the LTCN's architecture.

4.4.1 *LTCN's architecture*

The LTCN algorithm consists of two building blocks where the first one is designed to capture the dynamics of the system. This first building block, contains a LTCN model where each neuron maps a feature through an unsupervised learning approach. The second building block is designed to connect the inner neurons denoting the features with the decision neurons. Unlike the first neural block, the second one uses a supervised learning approach (Napoles et al., 2022).

The second component of the algorithm is a recurrence-aware sub-network that connects each temporal state with the decision neurons. This network uses the results of these states from the recurrent reasoning rule for a new instance. This is where the LTCN model differs from traditional FCM-based classifiers. LTCN considers all temporal states that are produced during the recurrent reasoning process, while the latter only considers the last state. This makes the LTCN model less sensitive to the unique fixed-point attractor (Napoles et al., 2022). The decision models of a classic FCM-based classifier and the LTCN algorithm are visualized in figure 5. Where 'A' indicates the temporal states, 'W' the inner weights connecting the features, 'B' the bias weights, and 'R' represents the outer weights connecting the temporal states with the decision neurons.



I). traditional FCM-based classifier

II). Recurrence-aware classifier

Figure 5: LTCN decision model

4.4.2 *LTCN Hyperparameter Tuning*

The LTCN documentation was consulted to determine the hyperparameter grid (see table 4). The method parameter specifies the regression approach and the function parameter specifies the activation function used when fitting the model. The 'T' parameter specifies the number of iterations to be performed. The 'Phi' parameter denotes the amount of non-linearity used during reasoning and finally, 'Alpha' specifies the positive penalization for L2-regularization (Napoles et al., 2022).

Table 4: The selected Hyperparameter grid for the LTCN model and the optimal value of the best-performing model after GridSearchCV

Parameters	Default Value	Parameter Grid	Best Value
method	'inverse'	['inverse', 'ridge']	'inverse'
T	20	[5, 10, 20, 40]	10
Function	'sigmoid'	['sigmoid', 'hyperbolic']	'sigmoid'
Phi	0.8	[0.2, 0.4, 0.6, 0.8, 1]	0.2
Alpha	1.0E-4	[0.00001, 0.0001, 0.001, 0.01]	0.00001

4.5 *SMOTE*

The dataset exhibits class imbalance, with 16% representing the minority 'Lethal complication' class and 84% the majority 'Alive' class. Imbalanced datasets tend to create a bias towards the majority class in predictive modeling, as highlighted by Blagus and Lusa (2013) (Blagus & Lusa, 2013). To enhance the model's ability to correctly predict the minority class, Synthetic Minority Oversampling Technique (SMOTE) has been employed to resample the training data.

Synthetic Minority Oversampling (SMOTE), a widely adopted resampling technique for imbalanced datasets, represents an enhanced method over random oversampling. The algorithm creates synthetic examples based on interpolation between several minority classes within a defined neighborhood (Fernández et al., 2018). Importantly, SMOTE is exclusively applied to the training data to ensure that the test set remains untouched for predictions post-model training. Additionally, a pipeline is created to apply SMOTE and avoid data leakage into the validation set. To asses the effect of SMOTE, model performances will be analyzed both with and without its application.

4.6 *Feature Importance Methods*

SHAP will be tested as one of the feature importance methods in this study, it assigns an importance value to each feature for a particular prediction. SHAP creates a theoretically reliable way of explaining model predictions by combining several existing methodologies to show how estimations change after specific features are removed. The magnitude of this change is quantified in SHAP values that can either be positive or negative (Lundberg & Lee, 2017; Scavuzzo et al., 2022). Besides SHAP, SKLearn's random feature permutation method will also be tested. This method captures the dependence of a model on a feature by randomly shuffling a single feature value. Finally, the built-in feature importance method of the best-performing algorithm will conclude the tested feature importance methods.

The performance of the three methods will be analyzed through the pixel flipping experiment, which tests the robustness of the feature importance methods. The term "pixel flipping" originates from its initial application in image interpretability, where it assesses the impact of changing specific pixels. In this study, the pixel flipping experiment will systematically marginalize the top features by imputing the mean value. This happens iteratively from the highest-ranked feature to the lowest, as determined by each feature importance method.

5 EXPERIMENTAL SETUP

5.1 *Data and pre-processing*

The anonymized dataset is publicly accessible in csv format through the UC Irvine machine learning repository and originates from the Krasnoyarsk Interdistrict Clinical Hospital in Russia (Dua & Graff, 2017). Comprising 111 distinct features and 1,700 rows, the dataset includes both categorical data like gender and numerical features such as blood pressure. The dataset is labeled with various complications of Myocardial Infarction (MI), with the lethal outcome designated as the target label for the binary classification task. The patient data can be divided into 2 classes: "Still Alive" (84%) and "Lethal Complication" (16%). These classes distinguish between patients who remain alive at the end of data collection and those who suffered a lethal complication after the Myocardial Infarction.

The dataset is rich in information and includes extensive descriptions for all its features. Notably, the dataset is clean and mostly complete which contributes to the task at hand. However, a class imbalance is present in the dataset, with considerably more patients belonging to the "Still Alive" class than the "Lethal Complication" class. The data originates from patients who were admitted to the hospital following an MI incident, with data collection at admission and the end of the first, second, and third day. For the purpose of this study, the focus is on the data collected at the time of admission since it is desirable to predict the medical outcomes of patients as soon as possible. This refinement reduced the dataset to 102 features and 1,700 instances.

5.1.1 *Missing Values*

Upon inspecting the missing values in the dataset, it was evident that 4 features were missing more than 60% of their data. Among these, two features pertained to systolic blood pressure, one measured the Creatine Phosphokinase value in the blood, and the last feature provided information about heredity to heart failure. Consequently, the final dataset was reduced to 98 features and 1,700 instances. To address the remaining missing values in features, the MICE imputation algorithm was employed, which fills missing values based on observed patterns in existing data (see section 5.2).

5.1.2 *Outlier Analysis*

An outlier analysis was conducted to find potential extreme values in the dataset. For the categorical features in the dataset, calling the describe function on the data frame provided a quick overview of the minimum and maximum values. For these features, no outliers were detected. For the numerical data features like age and blood pressure, histogram and boxplot distribution plots were plotted (see figure 6, 7, 8, 9). Although some more extreme values were observed, they were retained in the dataset as they remained within realistic ranges.

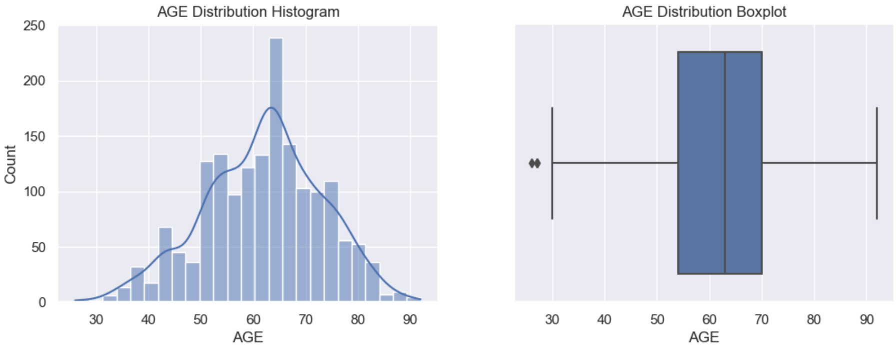


Figure 6: Age Outlier Analysis

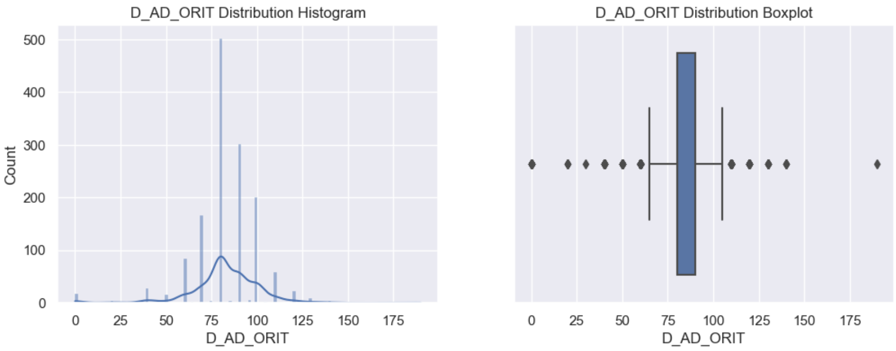


Figure 7: Diastolic BP Outlier Analysis

5.1.3 *Feature Correlation Analysis*

A feature correlation analysis provided insight into the correlation and relationships between features. The ten features exhibiting the highest correlation with the target variable were selected to create the correlation matrix. Since there are many features, it is impossible to provide a clear

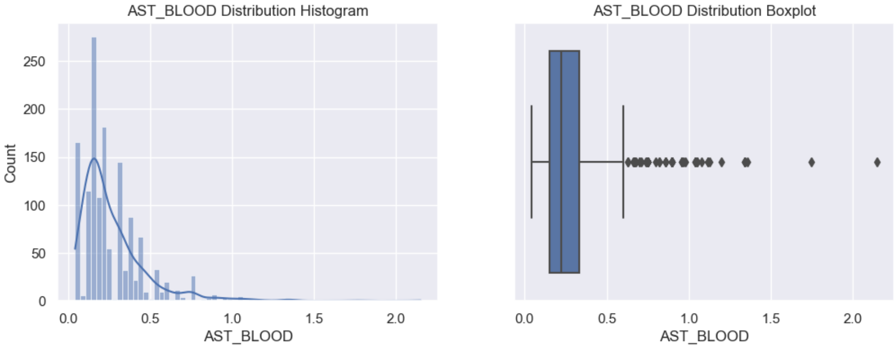


Figure 8: Serum AsAt content Outlier Analysis

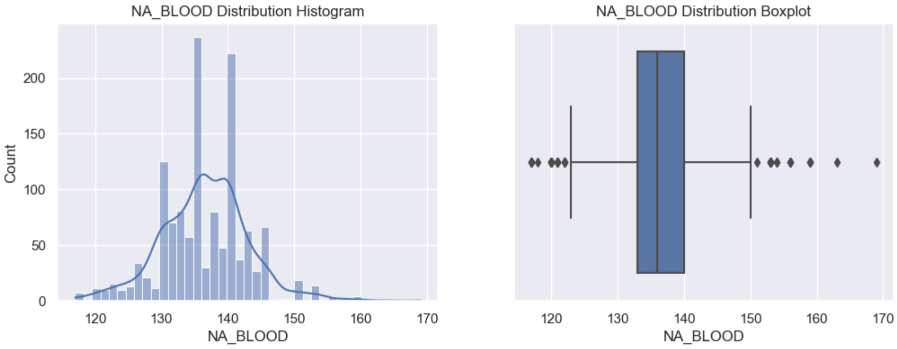


Figure 9: Serum Sodium content Outlier Analysis

correlation matrix featuring all features. After inspecting the correlation matrix, no problematic correlations were encountered.

5.2 *MICE Imputation*

Multivariate Imputation by Chained Equations (MICE) is a popular and widely employed method for handling missing values. This approach predicts missing values through iteratively running a series of predictive models. MICE assumes that data is missing at random, missing values are then predicted based on the data that is available. Generally, 5-10 iterations should be sufficient to obtain reliable estimates for the missing values (Azur et al., 2011). In this study, the number of iterations was set to 5, with 50 imputations per iteration for both the training and test sets.

5.3 *Train-Test Data Split*

To ensure the generalization ability of the models, the data will undergo a stratified train-test split. The data will be split, reserving 80% for the training set and 20% for the test set. The stratified split will be executed based on the target variable, ensuring that both sets maintain the same ratio of both majority and minority classes. Furthermore, during hyperparameter tuning, GridSearchCV will leverage the training set in combination with stratified k-fold cross-validation. The training set is split into multiple subsets of data, using one of the subsets as the test set while the model is trained on the remaining data. This process repeats iteratively and the model's performance is evaluated on the test set within each cross-validation iteration. Through this method, the optimal hyperparameters from the predetermined parameter grid are determined.

5.4 *Class Imbalance*

A class imbalance is present in the MI dataset used in this study. Around 84% of the patients belong to the class that is "Alive" and 16% of the patients belong to the "Lethal Complication" class. As inferred in section 3.3, there are various ways to address class imbalances. This study will apply SMOTE in a pipeline during GridsearchCV to explore the effect of SMOTE on model performances without data leakage into the test set.

5.5 *Hyperparameter Optimization*

Hyperparameter tuning is an essential step of the data science pipeline, as it can mitigate overfitting/underfitting, improve model performance, and help create more robust models. In this study, GridSearchCV was utilized to find the optimal hyperparameters for each model within a predetermined parameter grid. GridSearchCV was chosen as the more exhaustive method as opposed to RandomizedSearchCV. StratifiedKFold was utilized to ensure an equal distribution of each class within each fold during cross-validation and hyperparameter tuning (Pedregosa, Varoquaux, Gramfort, Michel, Thirion, Grisel, Blondel, Prettenhofer, Weiss, Dubourg, et al., 2011a). The predetermined hyperparameter grid for each model can be found in section 4. Each algorithm's documentation was consulted to obtain the parameter grids.

5.6 *Evaluation Strategy*

In this work, Cohen's Kappa score (Cohen, 1960) is considered the most important performance metric of each model. Kappa estimates the inter-rater agreement for categorical items and can range from -1 to 1. Within this range, -1 means there is no agreement and a value of 1 indicates a perfect agreement. A value of 0 indicates a random agreement. The kappa score provides a more robust measure for the evaluation of predictions on imbalanced datasets (Napoles et al., 2022).

Additionally, more traditional performance metrics like accuracy, recall, precision, F1, and ROC-AUC will be considered to provide comparability with previous studies. Finally, an error analysis will be provided for the model's ability to predict each class. This is especially desirable since there is a class imbalance present in the MI dataset. Providing an error analysis in the form of confusion matrices provides a more complete overview of model performance, unlike previous studies (Farah et al., 2022; Joshi et al., 2022; Reddy & Thangam, 2022).

5.7 *Description of Actual Implementation*

Python (version 3.11.5) (Van Rossum & Drake Jr, 1995) will be used as the programming language for this project. The Logistic Regression (Cox, 1958), XGBoost (Chen & Guestrin, 2016a), TabNet (Arik & Pfister, 2021), and LCTN (Napoles et al., 2022) algorithms will be used to generate binary classification models. Various packages were installed to use for various tasks within the data science pipeline: Numpy (Harris et al., 2020), Pandas

(McKinney et al., 2010), Imbalanced Learn (Lemaître et al., 2017), Matplotlib (Hunter, 2007), Seaborn (Waskom et al., 2017), Scikit-learn (Pedregosa, Varoquaux, Gramfort, Michel, Thirion, Grisel, Blondel, Prettenhofer, Weiss, Dubourg, et al., 2011b), Pytorch (Paszke et al., 2019), Miceforest, and Shap. Finally, Visual Studio Code (version 1.82.2) will be used as a code editing program.

6 RESULTS

This section will discuss the performance of the tested algorithms and their corresponding models. Cohen's Kappa score is used as the main metric to measure model performance. According to Cohen (1960), the kappa score should be interpreted as follows: "values lower or equal to 0 as indicating no agreement and 0.01–0.20 as none to slight, 0.21–0.40 as fair, 0.41– 0.60 as moderate, 0.61–0.80 as substantial, and 0.81–1.00 as an almost perfect agreement" (Cohen, 1960).
Additionally, accuracy, recall, precision, F1, and the ROC-AUC score will also be discussed. These performance metrics will provide comparability with previous studies. This section will also provide insights into the most prominent features and into the effect of resampling techniques to handle class imbalance. Lastly, confusion matrices will be evaluated to provide insight into the model's ability to classify each class.

6.1 *Overview Model Performances*

When looking at table 5, it is clear that the XGBoost + SMOTE model outperformed the other models with a kappa score of 0.572, an ROC-AUC of 0.764, and an overall test accuracy of 89.4%. Interestingly, without the application of SMOTE, the LTCN model had the superior performance over the other models with a kappa score of 0.542, an ROC-AUC score of 0.737, and an overall test accuracy of 89.1%. Furthermore, table 5 shows that SMOTE improved the Kappa and ROC-AUC scores for all models. The overall test accuracy however, decreased after SMOTE for every model except the XGBoost model. This indicates that SMOTE made the models more robust and better at correctly predicting the minority class. It is also important to note that the LTCN model + SMOTE provided the best ROC-AUC score of 0.820. Finally, it is fair to say that TabNet did not meet the expectations set by prior studies. The most likely explanation could be that the small dataset available for this study was not suitable for the complex TabNet algorithm. So to conclude, the XGBoost + SMOTE was the best-performing and most robust model with a kappa score of 0.572.

Table 5: Overview of the model performances

Metric	Kappa	ROC-AUC	Test Accuracy
Baseline model	0.347	0.658	84.1%
XGBoost	0.465	0.701	87.6%
TabNet	0.228	0.592	83.2%
LTCN	0.542	0.737	89.1%
Metric	Kappa	ROC-AUC	Test Accuracy
Baseline + SMOTE	0.393	0.760	78.5%
XGBoost + SMOTE	**0.572**	0.764	**89.4%**
TabNet + SMOTE	0.343	0.689	80.6%
LTCN + SMOTE	0.491	**0.820**	82.1%

Table 6 shows the precision, recall, and F1 score for the best-performing models and both classes. XGBoost and LTCN provide the best results. XGBoost + SMOTE does not necessarily provide the best score for each metric, but as is evident in table 5, it is the more robust model with the highest Kappa score. High scores for LTCN + SMOTE on the minority class are at the expense of the model's ability to predict the majority class correctly.

Table 6: Overview of the error analysis best-performing models. P = Precision, R = Recall

Class	'Alive'			'Lethal'		
Metric	P	R	F1	P	R	F1
Baseline + SMOTE	0.938	0.797	0.862	0.402	0.722	0.517
XGBoost + SMOTE	0.922	0.955	**0.938**	0.705	0.574	**0.633**
TabNet + SMOTE	0.904	0.860	0.882	0.412	0.519	0.459
LTCN - SMOTE	0.911	**0.965**	0.937	**0.737**	0.509	0.602
LTCN + SMOTE	**0.959**	0.821	0.885	0.469	**0.818**	0.596

6.2 *Baseline Logistic Regression Model*

The Logistic Regression algorithm was used to build a baseline model for this study. This baseline model provides a benchmark for the more advanced algorithms that are tested in this study. The hyperparameters were optimized using GridSearchCV and the model was tested with and without SMOTE to address the class imbalance. The results of the optimized baseline model can be seen in table 7.

Table 7: Performance metrics for Logistic Regression models

Metric	Kappa	ROC-AUC	Test Accuracy
Logistic Regression	0.347	0.658	**84.1%**
Logistic Regression + SMOTE	**0.393**	**0.760**	78.5%

Immediately, table 7 shows that it is important to include more performance metrics than only the accuracy score when dealing with an imbalanced dataset. While the more robust performance metrics improve after applying SMOTE to the training data, the overall model accuracy goes down. Since Kappa and the ROC-AUC score are considered more important measures of model performance in this study, we can conclude that SMOTE improves model performance. The best-performing baseline model provides a kappa score of 0.393 and an ROC-AUC score of 0.760.

6.2.1 *Logistic Regression error analysis*

When looking at the model's ability to predict each class, it is important to note that 16% of the patients belong to the "lethal complication" class, and 84% belong to the "Alive" class. In this error analysis, we only look at the best-performing baseline model. The confusion matrix for this model can be seen in figure 10.

Table 8: Overview of Recall, Precision and F1 score for the baseline model

Metric	Recall	Precision	F1-score
Lethal Complication	0.722	0.402	0.517
Alive	0.797	0.938	0.862

Upon inspecting table 8, it is evident that the recall scores for both classes are relatively close. However, the precision score exhibits a very high value for the majority class and a relatively low score for the minority class. Coupled with a lower F1-score, this suggests that the model encounters more difficulty in predicting the minority class and tends to produce more false positives.

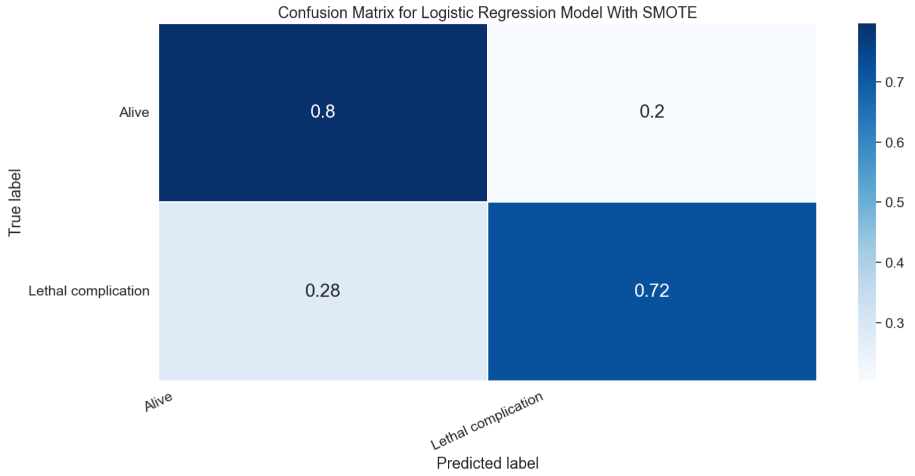


Figure 10: Confusion Matrix Logistic Regression with SMOTE

6.3 *XGBoost model*

In table 9 we can see that the XGBoost model in combination with SMOTE was the superior model since the kappa, ROC-AUC, and accuracy scores are all higher. A kappa score of 0.572 indicates there is moderate agreement, on the verge of substantial agreement between the raters. The overall accuracy of the best-performing model was 89.4%. In the next subsection, the error analysis of this model will be provided.

Table 9: Performance metrics for XGBoost models

Metric	Kappa	ROC-AUC	Test Accuracy
XGBoost	0.465	0.701	87.6%
XGBoost + SMOTE	**0.572**	**0.764**	**89.4%**

6.3.1 *XGBoost Error Analysis*

Upon inspecting table 10, a clear difference can be seen in the model's ability to predict each class. The model scores very highly for the majority "Alive" class, while it is unable to provide the same results for the minority class. A recall of 0.955 indicates that the model can correctly predict 95.5% of the "alive" class with a relatively high precision score of 0.922. When comparing the recall score for predicting the minority class of the XGBoost model with the Logistic Regression model, we can see that the latter provided a higher recall score. This can be explained by looking at the precision score, which indicates a considerably lower rate of false positives compared to the Logistic Regression model (table 10).

Table 10: Overview of Recall, Precision and F1 score for the best-performing XGBoost model

Metric	Recall	Precision	F1-score
Lethal Complication	0.574	0.705	0.633
Alive	0.955	0.922	0.938

The confusion matrix for the XGBoost model can be seen in figure 11, which gives a visual overview of the predictions made by the model.

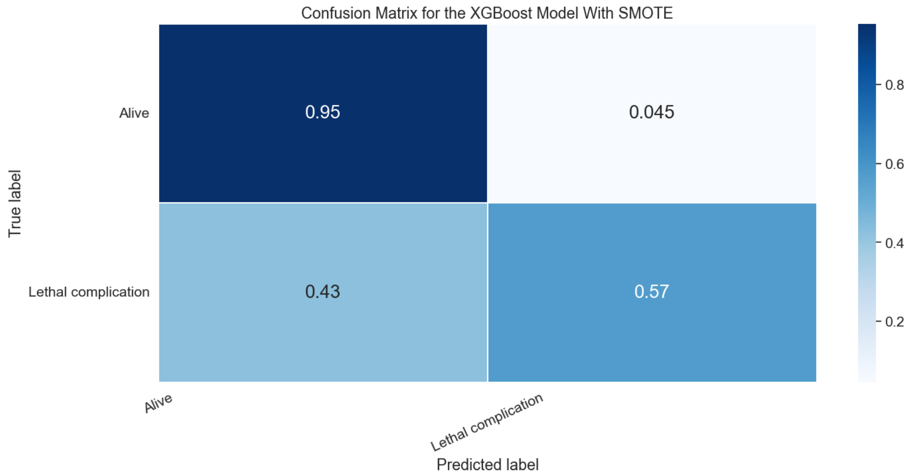


Figure 11: Confusion Matrix XGBoost with SMOTE

6.4 *TabNet Model*

The TabNet algorithm was selected based on the prior studies where TabNet was able to outperform XGBoost and other conventional machine learning algorithms on tabular data. Interestingly though, when looking at the kappa scores of 0.228 and 0.343 (see table 11), TabNet did not perform very well. It was outperformed by the baseline Logistic Regression model. This suggests that TabNet might not be suitable for this dataset and supports the implication that it works better with large and high dimensional datasets (Arik & Pfister, 2021).

When evaluating the effect of SMOTE in combination with the TabNet model, it is evident that SMOTE improves the kappa and ROC-AUC scores. The overall model accuracy, however, decreases after SMOTE is applied to

Table 11: Performance metrics for the TabNet models

Metric	Kappa	ROC-AUC	Test accuracy
TabNet	0.228	0.592	**83.2%**
TabNet + SMOTE	**0.343**	**0.689**	80.6%

the training data. This indicates that after applying SMOTE, the model's ability to correctly predict the minority class increases. Contrary to this, the decrease in the overall accuracy score, most likely indicates that this is at the expense of the model's ability to predict the majority class.

6.4.1 *TabNet Error Analysis*

Table 12: Overview of Recall, Precision and F1 score for the best-performing TabNet model

Metric	Recall	Precision	F1-score
Lethal Complication	0.519	0.412	0.459
Alive	0.860	0.904	0.882

Table 12 shows the recall, precision, and F1-score of the best-performing TabNet model. Upon inspecting these scores, it is evident that the model encounters the most difficulty in accurately predicting the minority 'lethal complication' class. This observation is further emphasized when reviewing the confusion matrix in figure 12.

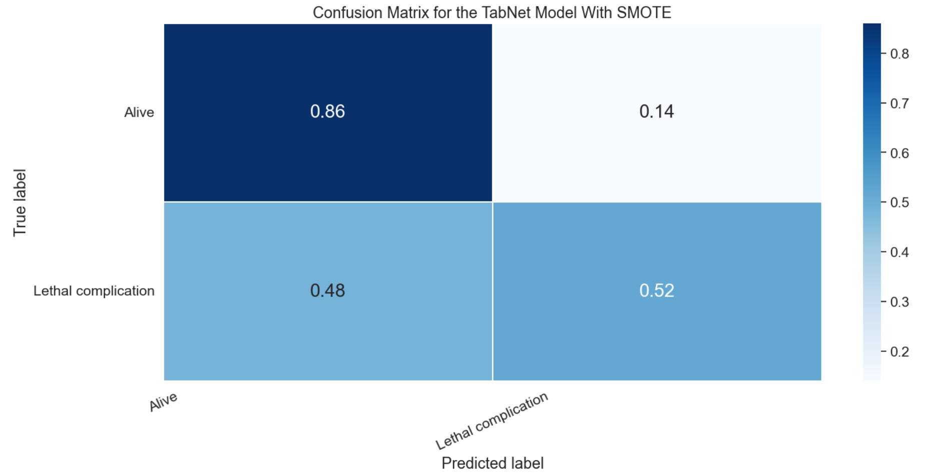


Figure 12: Confusion Matrix TabNet model with SMOTE

6.5 *Long-Term Cognitive Network*

The final algorithm assessed in this study is Long-Term Cognitive Network (LTCN), a deep learning model chosen for its potential to surpass traditional machine learning approaches on tabular data (Napoles et al., 2022). An interesting observation emerges when inspecting the results of the LTCN models: it demonstrates notable performance even without the application of SMOTE to address the class imbalance. The LTCN model without SMOTE yields a kappa score of 0.542, outperforming the model with SMOTE, which achieves a kappa score of 0.491. Given the emphasis on kappa as the primary performance metric in this study, the LTCN model without SMOTE is considered the best-performing LTCN model. It's noteworthy that the LTCN model in conjunction with SMOTE provided a relatively high ROC-AUC score of 0.820, while the overall model performance decreased by 7% after applying SMOTE to the training data (see table 13).

Table 13: Performance metrics for the LTCN models

Metric	Kappa	ROC-AUC	Test Accuracy
LTCN	**0.542**	0.737	**89.1%**
LTCN + SMOTE	0.491	**0.820**	82.1%

6.5.1 *LTCN Error Analysis*

Table 14 depicts the model's ability to correctly predict each class. Similar to the previously assessed models, LTCN encounters the most difficulty in accurately predicting the minority 'lethal complication' class. However, it exhibits a relatively high precision score, suggesting that the erroneous predictions have a lesser impact on the model's ability to accurately predict the majority 'alive' class. Figure 13 shows the confusion matrix for the LTCN model without SMOTE which provided the highest kappa score.

Table 14: Overview of Recall, Precision and F1 score for the best-performing LTCN model (highest Kappa score)

Metric	Recall	Precision	F1-score
Lethal Complication	0.509	0.737	0.602
Alive	0.965	0.911	0.937

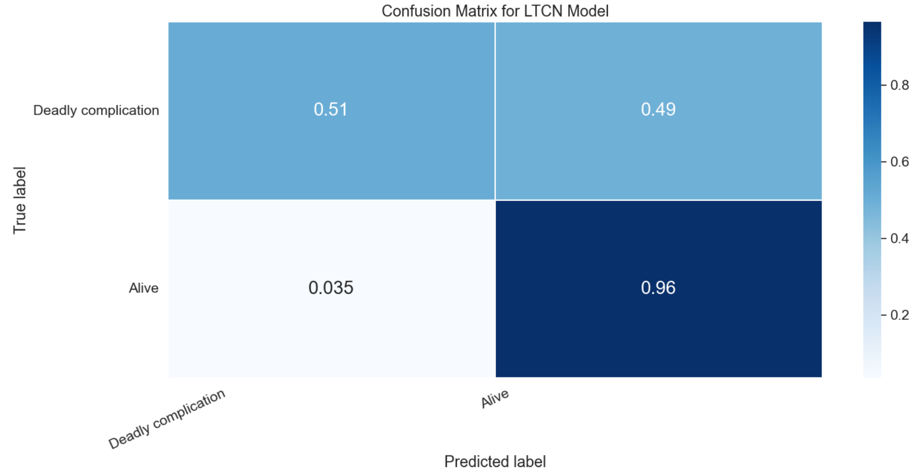


Figure 13: Confusion Matrix LTCN model without SMOTE

Table 15: Overview of Recall, Precision, and F1 score for the LTCN model with SMOTE (lower Kappa score)

Metric	Recall	Precision	F1-score
Lethal Complication	0.818	0.469	0.596
Alive	0.821	0.959	0.885

Considering the LTCN algorithm, the model with SMOTE was also analyzed and reported since it provided the highest overall ROC-AUC score. Upon comparing table 14 and 15, the relatively high recall score for the 'lethal complication' class is remarkable. On further inspection, this also comes with a significant drop in precision score, indicating that more false positives are predicted. Figure 14 shows the confusion matrix of the LTCN model with the highest ROC-AUC score. As can be seen, the correct predictions are balanced for both classes, and SMOTE significantly improves the recall score for the minority 'lethal complication' class.

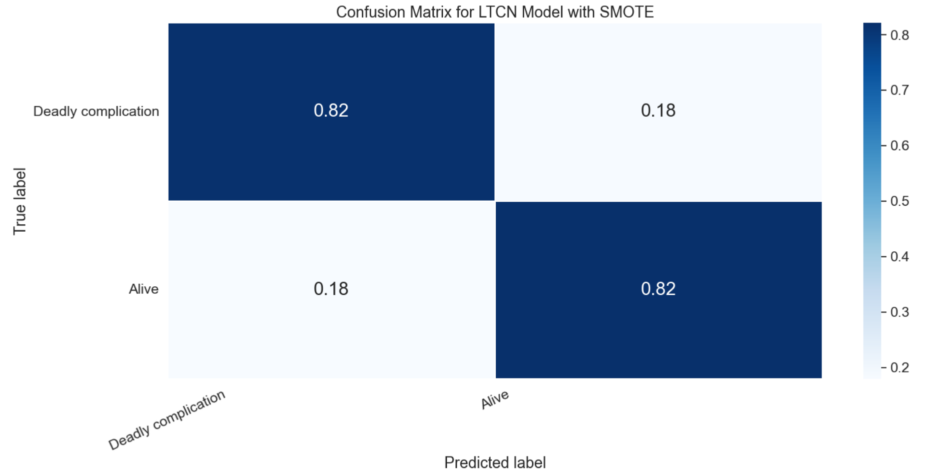


Figure 14: Confusion Matrix LTCN model with SMOTE

6.6 *Feature Importance Analysis*

To enhance the societal relevance of this study, a feature importance analysis was conducted. Identifying which features contribute most to the predictive power of the model can help determining which type of data should be collected. Three feature-importance methods were tested in a pixel-flipping experiment to find out which method provides the best insight into the most prominent features of the best-performing model. The three methods include Shap, XGBoost feature importance, and SKLearn's random feature permutation method. In the pixel-flipping experiment, the top feature is marginalized by imputing the mean value. This is done iteratively from top to bottom, based on the ranking made through each method. The result can be seen in figure 15.

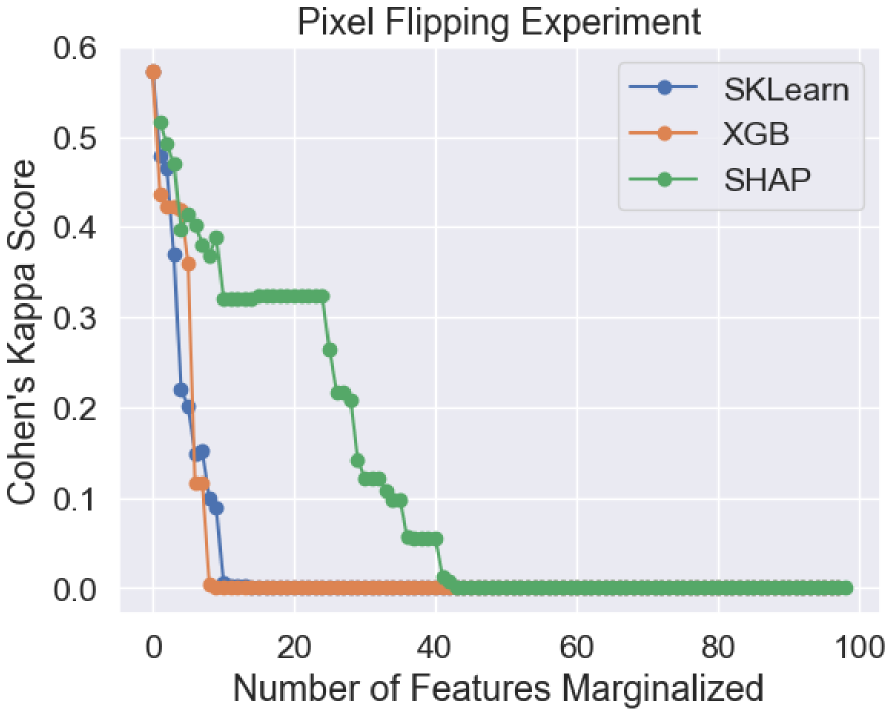


Figure 15: Pixel Flipping Experiment to identify best feature importance method

The graph shows that all three methods work quite well. In particular, the XGB and Sklearn methods seem to work well since the Kappa score decreases more or less every time the top feature is marginalized. Right around the 10 features mark, the kappa score is zero or lower. This implies that after the top 10 features have been marginalized, the model's predictive power is at less than chance level. Thus, also showing the impact of these top 10 features and their importance for the classification problem in this study. Based on this analysis, the XGBoost feature importance method will identify the most prominent features (see figure 16).

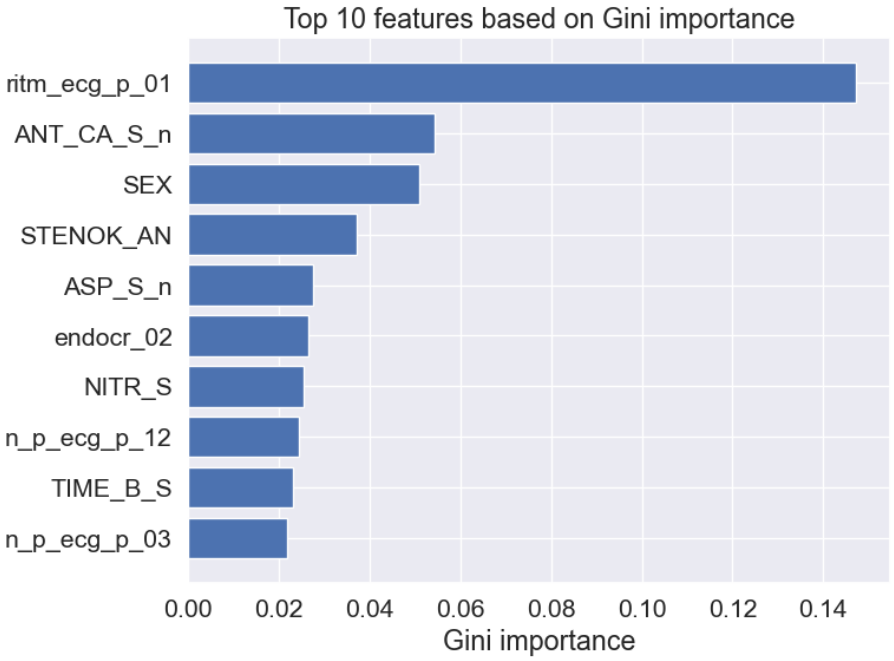


Figure 16: XGBoost top 10 Gini feature importance ranking

The most prominent feature according to the Gini importance ranking is the ECG rhythm at the time of admission to the hospital, which checked patients for a healthy heart rate between 60-90. The second most prominent feature checked if calcium channel blockers were used in the ICU. The sex of the patient was identified as the third most prominent feature. Followed by the fourth feature that checked patients when they last experienced chest pain. The fifth most prominent feature provides information about the Aspirin intake in the ICU. The sixth most prominent feature provided information about the Diabetes status of the patients. The seventh most prominent feature provided information about the use of liquid nitrates in the ICU. Followed by the eight most prominent feature that provides information on the presence of a complete Right Bundle Branch Block within the heart. The ninth most prominent feature provides information about the time elapsed from the beginning of the heart attack to admission into the hospital. Finally, the last feature of the ranking provides information on the presence of a first-degree atrioventricular (AV) block. The description of the feature names seen in figure 16, can be found in appendix A.

7 DISCUSSION

7.1 *General Results*

This thesis aims to contribute to the limited literature in the field of MI complication mortality classification in combination with deep learning techniques for tabular data. Notably, it represents the first instance of evaluating the Long-Term Cognitive Network (LTCN) algorithm for this classification problem. Additionally, regarding the TabNet algorithm, it is the first study in this in this domain that tests TabNet on a relatively small dataset. In the end, four algorithms were tested. Logistic Regression was used to build the baseline model and XGBoost was used to build the machine learning model since it is widely considered to be the current state-of-the-art for predictive modeling on tabular data. The class imbalance in the dataset was addressed by applying SMOTE to the training data.

After comparing the model performances, it was clear that XGBoost and LTCN performed the best. Interestingly, TabNet did not meet the expectations set by the performance it showed in several previous studies (de Carvalho et al., 2023; Kim et al., 2023; Nguyen & Byeon, 2023). When looking at the kappa score of 0.343, it becomes apparent that TabNet was even outperformed by the baseline model's score of 0.393. One explanation for this could be the relatively small dataset that was available for this classification problem. Since TabNet is a complex model that is designed for high dimensional and large datasets (Arik & Pfister, 2021). Still, it was expected to outperform the baseline model and provide a good comparative model to the LTCN and XGBoost models.

The LTCN and XGBoost models performed better than TabNet and they provided competitive scores. The XGBoost model with SMOTE provided a kappa score of 0.572 while the LTCN model without SMOTE provided a kappa score of 0.542, which is only a 0.03 difference. Interestingly, LTCN managed to score this kappa score without the application of SMOTE. XGBoost only had a kappa score of 0.465 when SMOTE was not applied. Furthermore, the LTCN model with SMOTE provided the highest overall ROC-AUC score of 0.820. XGBoost's highest ROC-AUC score was 0.764. When comparing these two algorithms, it is fair to say that they are competitive in this study. LTCN and XGBoost can be seen as suitable algorithms for the classification problem in this study, unlike TabNet.

7.2 *Related Works*

When comparing this work to the previous studies in the field of MI complications and artificial intelligence similarities and differences can be seen. First of all, distinctions can be made in the goal of the study. Some studies aimed to build machine learning models to predict the likelihood of a Myocardial Infarction (Kora & Sri Rama Krishna, 2016; Kora et al., 2018; Sharma et al., 2018). Others tried to predict the long-term survival rate after an infarction (Nguyen & Byeon, 2023). While there were also studies that aimed to predict the short-term outlook after an infarction (Farah et al., 2022; Joshi et al., 2022; Newaz et al., 2023; Reddy & Thangam, 2022). A similarity across most studies is the imbalanced datasets. Which has been treated in various ways, like varying resampling methods and adding a higher cost function to erroneous predictions of the minority class (Newaz et al., 2023). An important aspect missing in the previous studies on the same dataset, is the missing error analysis to provide transparency of the model's limitations. The difference in input data presents another difference, notably some studies like de Carvalho et al. (2023)'s study also included data information like smoking habits. Finally, the size of the datasets of studies differ, which as can be seen from this work, is an important consideration when choosing an algorithm to work with.

Prior studies that have worked on building predictive models on the same dataset as this study can provide a good benchmark (Farah et al., 2022; Joshi et al., 2022; Newaz et al., 2023; Reddy & Thangam, 2022). The results in these prior studies provide comparability and validity for the performance of the models in this work. Newaz et al. (2023) provided a concise overview of the results of the models tested in the previous studies on this dataset. This overview will be used to compare with the model performances in this work (see figure 16).

Table 16: Comparing results with previous studies

Studies	**Accuracy**	**ROC-AUC**
Farah	87.3	61.7
Reddy	84.9	68.0
Joshi	86.6	57.2
Newaz	**91.9**	80.9
LTCN + SMOTE	82.1	**82.0**
LTCN - SMOTE	89.1	73.7
XGBoost + SMOTE	89.4	76.4

The best-performing models in this work outperformed most of the models tested in prior studies on this dataset. It is important to note that the best performing model in Newaz et al. (2023)'s work used XGBoost and a newly proposed combination of methods to treat class imbalance. The LTCN algorithm in our work managed to outperform most conventional machine learning algorithms tested in previous studies and even scored the highest ROC-AUC score of 82.0. It is the first deep learning algorithm that has successfully provided competitive results on this data. It is fair to say that the algorithm is suitable for this classification problem. The relatively high accuracy score of 89.1 and ROC-AUC score of 73.7 without any treatment to the class imbalance is quite remarkable. Furthermore, LTCN seems to be more suitable for smaller datasets than the TabNet algorithm. Finally, it could be interesting to test the algorithm on more imbalanced datasets to see if it is fair to say that the algorithm works well on imbalanced datasets without treating class imbalance.

7.3 *Societal and Scientific impact*

> *"RQ1: To what extent do the performance metrics of the TabNet, LTCN, and XGBoost models compare to the performance metrics of the baseline model for predicting MI complication mortality? "*

Through RQ1 the model performances of TabNet, LTCN, and XGBoost have been compared with each other and the baseline Logistic Regression model. The main goal was to explore the potential of deep learning algorithms for tabular data and compare their performance with the current state-of-the-art (XGBoost) and a baseline model. To answer RQ1, the XGBoost + SMOTE model provided the best kappa score of 0.572. The LTCN provided relatively good results and was close to outperforming XGBoost. The TabNet model did not seem suitable for the classification problem since it was unable to outperform even the baseline model. While LTCN did not outperform XGBoost, it can be considered a suitable model for the classification problem. It could be possible that LTCN can outperform XGBoost in similar studies with similar classification problems.

> *"RQ2: To what extent does the model performance differ when using SMOTE to address the class imbalance in the dataset?"*

Considering RQ2, the effect of SMOTE on the models is clearly visible. With SMOTE applied to the training data, all models provide a higher ROC-AUC score, indicating that the models improved at correctly predicting the minority 'lethal complication' class. When looking at the kappa score, however, all models improved after SMOTE except for LTCN. Surprisingly, LTCN provided a high kappa score without SMOTE, achieving a score of 0.542. SMOTE did drastically improve the LTCN recall score for the minority class, which resulted in the highest ROC-AUC score of 0.820. Overall model accuracy decreased for all models except XGBoost after SMOTE was applied.

> *"RQ3: Which important features in the dataset can be identified through feature importance methods for the best-performing model?"*

Finally, RQ3 aimed to identify the most prominent features for the best-performing model. Identifying these features with the most predictive power can provide insight into the valuable data that should be collected to create successful predictive models. After testing three methods and doing a pixel-flipping experiment to decide on the best method, XGBoost feature importance ranking was used to answer RQ3. The three most prominent features were: The ECG rhythm at the time of admission to the hospital, The use of calcium blockers in the ICU, and the sex of the patient.

So to conclude, the approach in this work has identified a new deep learning algorithm that is suitable for the classification task and could outperform the current state-of-the-art XGBoost algorithm in similar classification tasks. This contributes to the shift from machine to deep learning that was identified in the medical field. Additionally, it interestingly provides a relatively good performance of LTCN without class imbalance treatment that could be explored further in future research. Furthermore, this study provides insight into the TabNet algorithm and its difficulty with relatively small datasets. Lastly, this study gives a transparent overview of the limitations of the tested models, unlike prior studies.

7.4 *Limitations and Future Work*

Some limitations have been identified in this study. The first one being, more resampling techniques could have been tested to see the effect on the models. Since the main difference with the best-performing model of Newaz et al. (2023)'s study is the imbalanced data treatment, where a newly proposed combination of methods was used. Future research could explore the best resampling techniques for deep learning algorithms in binary classification tasks, since the LTCN model was pretty sensitive to SMOTE which decreased model robustness. The unsuitability of the TabNet algorithm for the available dataset, presents the second limitation of this work. It would be interesting to test the TabNet algorithm on a much larger dataset in future work.

A third limitation is the model performances when it comes to correctly predicting the minority 'lethal complication' class. To create a successful predictive model that can be applied in real life, it is crucial that patients at risk of dying are not misclassified. A main solution for this could be gathering a bigger dataset with more minority examples. Additionally, future work could test the LTCN algorithm on larger datasets to test it's performance on larger datasets. In this regard, it would be interesting to compare the performance of LTCN and TabNet on a larger and high-dimensional dataset within the research domain. Finally, upon comparison with other algorithms, the LTCN algorithm performed surprisingly well without any class imbalance treatment. Future research could test these algorithms on more imbalanced datasets to see if this hypothesis is true.

8 CONCLUSION

This research aimed to contribute to the existing literature for Myocardial Infarction (MI) complication mortality classification. This study sought to successfully create predictive models using recent deep learning algorithms explicitly designed for tabular data. The Long-Term Cognitive Network (LTCN) and TabNet algorithms were compared to the current state-of-the-art XGBoost algorithm. The XGBoost in conjunction with SMOTE model yielded the highest kappa score of 0.572, closely followed by the LTCN model without SMOTE, achieving a score of 0.542. Surprisingly, the TabNet algorithm was unable to outperform the baseline Logistic Regression model. This work can hopefully contribute to the advancements of deep learning techniques for tabular data in the medical field. Given that tabular data is still the biggest form of data, this will be crucial to keep up with the shift from machine to deep learning in the medical field.

REFERENCES

Abd Elrahman, S. M., & Abraham, A. (2013). A review of class imbalance problem. *Journal of Network and Innovative Computing*, *1*(2013), 332–340.

Alabi, R. O., Elmusrati, M., Leivo, I., Almangush, A., & Mäkitie, A. A. (2023). Machine learning explainability in nasopharyngeal cancer survival using lime and shap. *Scientific Reports*, *13*(1), 8984.

Arik, S. Ö., & Pfister, T. (2021). Tabnet: Attentive interpretable tabular learning. *Proceedings of the AAAI conference on artificial intelligence*, *35*(8), 6679–6687.

Austin, P. C., & Tu, J. V. (2004). Automated variable selection methods for logistic regression produced unstable models for predicting acute myocardial infarction mortality. *Journal of clinical epidemiology*, *57*(11), 1138–1146.

Azur, M. J., Stuart, E. A., Frangakis, C., & Leaf, P. J. (2011). Multiple imputation by chained equations: What is it and how does it work? *International journal of methods in psychiatric research*, *20*(1), 40–49.

Blagus, R., & Lusa, L. (2013). Smote for high-dimensional class-imbalanced data. *BMC bioinformatics*, *14*, 1–16.

Chakraborty, A., Chatterjee, S., Majumder, K., Shaw, R. N., & Ghosh, A. (2022). A comparative study of myocardial infarction detection from ecg data using machine learning. *Advanced Computing and Intelligent Technologies: Proceedings of ICACIT 2021*, 257–267.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). Smote: Synthetic minority over-sampling technique. *Journal of artificial intelligence research*, *16*, 321–357.

Chen, T., & Guestrin, C. (2016a). XGBoost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794. https://doi.org/10.1145/2939672.2939785

Chen, T., & Guestrin, C. (2016b). Xgboost: A scalable tree boosting system. *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 785–794.

Cho, S. M., Austin, P. C., Ross, H. J., Abdel-Qadir, H., Chicco, D., Tomlinson, G., Taheri, C., Foroutan, F., Lawler, P. R., Billia, F., et al. (2021). Machine learning compared with conventional statistical models for predicting myocardial infarction readmission and mortality: A systematic review. *Canadian Journal of Cardiology*, *37*(8), 1207–1214.

Cleveland. (2023). https://my.clevelandclinic.org/health/diseases/16818-heart-attack-myocardial-infarction

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and psychological measurement*, *20*(1), 37–46.

Cox, D. R. (1958). The regression analysis of binary sequences. *Journal of the Royal Statistical Society: Series B (Methodological)*, *20*(2), 215–232.

de Carvalho, L. S. F., Alexim, G., Nogueira, A. C. C., Fernandez, M. D., Rezende, T. B., Avila, S., Reis, R. T. B., Soares, A. A. M., & Sposito, A. C. (2023). The framing of time-dependent machine learning models improves risk estimation among young individuals with acute coronary syndromes. *Scientific Reports*, *13*(1), 1021.

Dua, D., & Graff, C. (2017). Uci machine learning repository. http://archive.ics.uci.edu/ml

Farah, C., Adla, Y. A., & Awad, M. (2022). Can machine learning predict mortality in myocardial infarction patients within several hours of hospitalization? a comparative analysis. *2022 IEEE 21st Mediterranean Electrotechnical Conference (MELECON)*, 1135–1140.

Fernández, A., Garcia, S., Herrera, F., & Chawla, N. V. (2018). Smote for learning from imbalanced data: Progress and challenges, marking the 15-year anniversary. *Journal of artificial intelligence research*, *61*, 863–905.

Garg, A., & Mago, V. (2021). Role of machine learning in medical research: A survey. *Computer science review*, *40*, 100370.

Gehring, J., Auli, M., Grangier, D., Yarats, D., & Dauphin, Y. N. (2017). Convolutional sequence to sequence learning. *International conference on machine learning*, 1243–1252.

Golovenkin, S. E., Bac, J., Chervov, A., Mirkes, E. M., Orlova, Y. V., Barillot, E., Gorban, A. N., & Zinovyev, A. (2020). Trajectories, bifurcations, and pseudo-time in large clinical datasets: Applications to myocardial infarction and diabetes data. *GigaScience*, *9*(11), giaa128.

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., Fernández del Río, J., Wiebe, M., Peterson, P., . . . Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*, 357–362. https://doi.org/10.1038/s41586-020-2649-2

Hunter, J. D. (2007). Matplotlib: A 2d graphics environment. *Computing in science & engineering*, *9*(3), 90–95.

Jang, H.-J., & Cho, K.-O. (2019). Applications of deep learning for the analysis of medical data. *Archives of pharmacal research*, *42*, 492–504.

Joshi, A., Gunwant, H., Sharma, M., & Chaudhary, V. (2022). Early prognosis of acute myocardial infarction using machine learning techniques. *Proceedings of Data Analytics and Management: ICDAM 2021, Volume 2*, 815–829.

Kim, T., Kim, M., Lee, H. W., & Song, G. (2023). One year mortality prediction in heart failure using feature selection and missing value imputation in deep learning. *2023 IEEE International Conference on Big Data and Smart Computing (BigComp)*, 145–148.

Knapič, S., Malhi, A., Saluja, R., & Främling, K. (2021). Explainable artificial intelligence for human decision support system in the medical domain. *Machine Learning and Knowledge Extraction*, *3*(3), 740–770.

Kora, P., Annavarapu, A., & Borra, S. (2018). Ecg based myocardial infarction detection using different classification techniques. *Classification in BioApps: Automation of Decision Making*, 57–77.

Kora, P., & Sri Rama Krishna, K. (2016). Ecg based heart arrhythmia detection using wavelet coherence and bat algorithm. *Sensing and Imaging*, *17*, 1–16.

Lai, V., Cai, J. Z., & Tan, C. (2019). Many faces of feature importance: Comparing built-in and post-hoc feature importance in text classification. *arXiv preprint arXiv:1910.08534*.

Lemaître, G., Nogueira, F., & Aridas, C. K. (2017). Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of Machine Learning Research*, *18*(17), 1–5. http://jmlr.org/papers/v18/16-365.html

Lewin-Epstein, O., Baruch, S., Hadany, L., Stein, G. Y., & Obolski, U. (2021). Predicting antibiotic resistance in hospitalized patients by applying machine learning to electronic medical records. *Clinical Infectious Diseases*, *72*(11), e848–e855.

Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Advances in neural information processing systems*, *30*.

McKinney, W., et al. (2010). Data structures for statistical computing in python. *Proceedings of the 9th Python in Science Conference*, *445*, 51–56.

Mohanty, A., & Mishra, S. (2022). A comprehensive study of explainable artificial intelligence in healthcare. In *Augmented intelligence in healthcare: A pragmatic and integrated analysis* (pp. 475–502). Springer.

Napoles, G., Salgueiro, Y., Grau, I., & Espinosa, M. L. (2022). Recurrence-aware long-term cognitive network for explainable pattern classification. *IEEE transactions on cybernetics*.

Nápoles, G., Salgueiro, Y., Grau, I., & Espinosa, M. L. (2021). Recurrence-aware long-term cognitive network for explainable pattern classification. *arXiv preprint arXiv:2107.03423*.

Newaz, A., Mohosheu, M. S., & Al Noman, M. A. (2023). Predicting complications of myocardial infarction within several hours of hospitalization using data mining techniques. *Informatics in Medicine Unlocked*, *42*, 101361.

Nguyen, H. V., & Byeon, H. (2023). Prediction of out-of-hospital cardiac arrest survival outcomes using a hybrid agnostic explanation tabnet model. *Mathematics*, *11*(9), 2030.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., . . . Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems 32* (pp. 8024–8035). Curran Associates, Inc. http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011a). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011b). Scikit-learn: Machine learning in python. *Journal of machine learning research*, *12*(Oct), 2825–2830.

Prendin, F., Pavan, J., Cappon, G., Del Favero, S., Sparacino, G., & Facchinetti, A. (2023). The importance of interpreting machine learning models for blood glucose prediction in diabetes: An analysis using shap. *Scientific reports*, *13*(1), 16865.

Putatunda, S., & Rama, K. (2018). A comparative analysis of hyperopt as against other approaches for hyper-parameter optimization of xgboost. *Proceedings of the 2018 international conference on signal processing and machine learning*, 6–10.

Rai, H. M., & Chatterjee, K. (2022). Hybrid cnn-lstm deep learning model and ensemble technique for automatic detection of myocardial infarction using big ecg data. *Applied Intelligence*, *52*(5), 5366–5384.

Razzak, M. I., Naz, S., & Zaib, A. (2018). Deep learning for medical image processing: Overview, challenges and the future. *Classification in BioApps: Automation of Decision Making*, 323–350.

Reddy, L., & Thangam, S. (2022). Predicting relapse of the myocardial infarction in hospitalized patients. *2022 3rd International Conference for Emerging Technology (INCET)*, 1–7.

Saarela, M., & Jauhiainen, S. (2021). Comparison of feature importance measures as explanations for classification models. *SN Applied Sciences*, *3*, 1–12.

Scavuzzo, C. M., Scavuzzo, J. M., Campero, M. N., Anegagrie, M., Aramendia, A. A., Benito, A., & Periago, V. (2022). Feature importance:

Opening a soil-transmitted helminth machine learning model via shap. *Infectious Disease Modelling*, *7*(1), 262–276.

Seki, T., Kawazoe, Y., & Ohe, K. (2021). Machine learning-based prediction of in-hospital mortality using admission laboratory data: A retrospective, single-site study using electronic health record data. *PloS one*, *16*(2), e0246640.

Sharifani, K., & Amini, M. (2023). Machine learning and deep learning: A review of methods and applications. *World Information Technology and Engineering Journal*, *10*(07), 3897–3904.

Sharma, M., San Tan, R., & Acharya, U. R. (2018). A novel automated diagnostic system for classification of myocardial infarction ecg signals using an optimal biorthogonal filter bank. *Computers in biology and medicine*, *102*, 341–356.

Shwartz-Ziv, R., & Armon, A. (2022). Tabular data: Deep learning is not all you need. *Information Fusion*, *81*, 84–90.

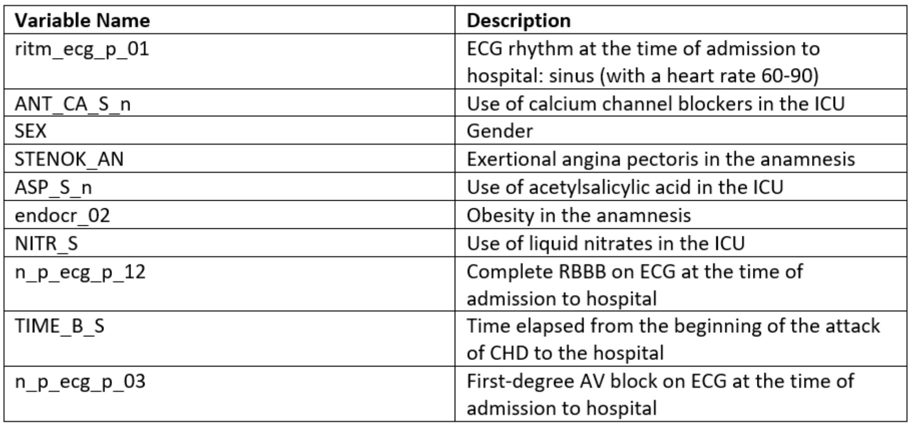
Tsien, C. L., Fraser, H. S., Long, W. J., & Kennedy, R. L. (1998). Using classification tree and logistic regression methods to diagnose myocardial infarction. In *Medinfo'98* (pp. 493–497). IOS Press.

Van Rossum, G., & Drake Jr, F. L. (1995). *Python reference manual*. Centrum voor Wiskunde en Informatica Amsterdam.

Waskom, M., Botvinnik, O., O'Kane, D., Hobson, P., Lukauskas, S., Gemperline, D. C., Augspurger, T., Halchenko, Y., Cole, J. B., Warmenhoven, J., de Ruiter, J., Pye, C., Hoyer, S., Vanderplas, J., Villalba, S., Kunter, G., Quintero, E., Bachant, P., Martin, M., . . . Qalieh, A. (2017, September). *Mwaskom/seaborn: V0.8.1 (september 2017)* (Version v0.8.1). Zenodo. https://doi.org/10.5281/zenodo.883859

Wu, Y., Zhang, L., Bhatti, U. A., & Huang, M. (2023). Interpretable machine learning for personalized medical recommendations: A lime-based approach. *Diagnostics*, *13*(16), 2681.

APPENDIX A



Variable Name	Description
ritm_ecg_p_01	ECG rhythm at the time of admission to hospital: sinus (with a heart rate 60-90)
ANT_CA_S_n	Use of calcium channel blockers in the ICU
SEX	Gender
STENOK_AN	Exertional angina pectoris in the anamnesis
ASP_S_n	Use of acetylsalicylic acid in the ICU
endocr_02	Obesity in the anamnesis
NITR_S	Use of liquid nitrates in the ICU
n_p_ecg_p_12	Complete RBBB on ECG at the time of admission to hospital
TIME_B_S	Time elapsed from the beginning of the attack of CHD to the hospital
n_p_ecg_p_03	First-degree AV block on ECG at the time of admission to hospital

Figure 17: Description of top 10 features